To prove by induction that it takes exactly F_n recursive calls to `fibonacci(1)` when computing `fibonacci(n)`, where F_n is the n-th Fibonacci number, we need to analyze the recursive structure of the Fibonacci sequence and establish the relationship between the number of calls to `fibonacci(1)` and the Fibonacci numbers.

Fibonacci Function Analysis

The given `fibonacci` method is a direct recursive implementation of the Fibonacci sequence:

```
public static long fibonacci(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;
    return fibonacci(n-1) + fibonacci(n-2);
}
```

The Fibonacci numbers are defined as: $F_0 = 0$ $F_1 = 1$ $F_n = F_{n-1} + F_{n-2} \quad \text{for} \quad n \geq 2$

Induction Proof

Base Cases

1. For $n = 0$:
   - `fibonacci(0)` directly returns 0 without making any recursive calls.
   - Therefore, there are 0 calls to `fibonacci(1)`.
   - $F_0 = 0$, which matches the number of calls.
2. For $n = 1$:
   - `fibonacci(1)` directly returns 1 without making any recursive calls.
   - Therefore, there is 1 call to `fibonacci(1)`.
   - $F_1 = 1$, which matches the number of calls.

These base cases hold true.

Inductive Step

Assume that for some $k \geq 1$, the number of recursive calls to `fibonacci(1)` when computing `fibonacci(k)` is exactly F_k.

We need to show that the number of recursive calls to `fibonacci(1)` when computing `fibonacci(k+1)` is exactly F_{k+1}.

When computing `fibonacci(k+1)`, the function makes the following calls:

- `fibonacci(k)`
- `fibonacci(k-1)`

According to the inductive hypothesis:

- The number of calls to `fibonacci(1)` in `fibonacci(k)` is F_k.
- The number of calls to `fibonacci(1)` in `fibonacci(k-1)` is F_{k-1}.

Therefore, the total number of calls to `fibonacci(1)` when computing `fibonacci(k+1)` is: $\text{Calls}(k+1) = \text{Calls}(k) + \text{Calls}(k-1)$
$\text{Calls}(k+1) = F_k + F_{k-1}$

By the definition of the Fibonacci sequence: $F_{k+1} = F_k + F_{k-1}$

Thus: $\text{Calls}(k+1) = F_{k+1}$

This completes the inductive step, showing that the number of calls to `fibonacci(1)` when computing `fibonacci(n)` is exactly F_n.

Conclusion

By mathematical induction, we have shown that it takes exactly F_n recursive calls to `fibonacci(1)` when computing `fibonacci(n)`, where F_n is the n-th Fibonacci number. This completes the proof.